



# **Document technique sur l'installation d'un serveur WEB**

**Par**

**Grégoire LE BARON**

## **Table des matières :**

**1- Introduction**

**2- Prérequis**

**3- Installation**

**4- Fonctionnement**

- a. Création de page HTML**
- b. Connexion à distance**

## 1- Introduction

Le serveur sert à héberger un site web qui sera visible pour présenter l'entreprise depuis n'importe quel navigateur.

## 2- Prérequis

Nous allons voir comment réaliser un serveur Web sur une machine virtuelle Debian. Une VM Debian a été installée en amont (cf. la documentation technique concernant l'installation de VM).

Sur le site de Debian, nous retrouvons le tableau suivant qui nous informe sur la configuration matérielle minimale recommandée.

| Type d'installation | RAM (minimum)  | RAM (recommandée) | Disque dur    |
|---------------------|----------------|-------------------|---------------|
| Sans bureau         | 128 mégaoctets | 512 mégaoctets    | 2 gigaoctets  |
| Avec bureau         | 256 mégaoctets | 1 gigaoctet       | 10 gigaoctets |

## 3- Installation

Dans un premier temps, nous devons mettre à jour la VM grâce à la commande : « **apt -y update && apt -y upgrade** ». « update » permet de mettre à jour les fichiers sources et « upgrade » les paquets.

Les mises à jour sont terminées lorsque l'écran propose d'entrer une nouvelle ligne de commande.

```
root@debian:/home/user# apt -y update && apt -y upgrade
Réception de :1 http://security.debian.org/debian-security bullseye-security InRelease [48,4 kB]
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Réception de :3 http://deb.debian.org/debian bullseye-updates InRelease [44,1 kB]
Réception de :4 http://security.debian.org/debian-security bullseye-security/main Sources [191 kB]
Réception de :5 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [284 kB]
Réception de :6 http://security.debian.org/debian-security bullseye-security/main Translation-en [158 kB]
570 ko réceptionnés en 1s (1 216 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:/home/user#
```

Maintenant que les mises à jour sont faites, nous devons procéder à l'installation des paquets grâce à la commande suivante « **apt -y install apache2 php mariadb-server libapache2-mod-php php-mysql** ».

- Apache2 est un serveur web, il sera chargé de répondre aux requêtes des utilisateurs souhaitant accéder au site.
- Le paquet php sera chargé d'interpréter les scripts PHP.
- libapache2-mod-php permet à apache d'utiliser PHP, et donc d'interpréter le code PHP directement depuis apache.
- mysql-server est un serveur de base de données, malgré le nom du paquet, mysql a été remplacé par MariaDB, l'ancien nom est resté pour des raisons de compatibilité.
- php-mysql permet à vos scripts PHP de pouvoir communiquer avec le serveur MySQL via un driver de connexion (ou ODBC).

L'écran indiquera la progression de l'installation :

```
Paramétrage de mariadb-client-core-10.5 (1:10.5.18-0+deb11u1) ...
Paramétrage de mariadb-client-10.5 (1:10.5.18-0+deb11u1) ...
Paramétrage de libhtml-parser-perl (3.75-1+b1) ...
Paramétrage de php7.4-opcache (7.4.33-1+deb11u3) ...

Creating config file /etc/php/7.4/mods-available/opcache.ini with new version
Paramétrage de libhttp-message-perl (6.28-1) ...
Paramétrage de apache2-utils (2.4.54-1~deb11u1) ...
Paramétrage de php7.4-json (7.4.33-1+deb11u3) ...

Creating config file /etc/php/7.4/mods-available/json.ini with new version
Paramétrage de php-mysql (2:7.4+76) ...
Paramétrage de libcgi-pm-perl (4.51-1) ...
Paramétrage de php7.4-cli (7.4.33-1+deb11u3) ...
update-alternatives: utilisation de « /usr/bin/php7.4 » pour fournir « /usr/bin/php » (php) en mode automatique
update-alternatives: utilisation de « /usr/bin/phar7.4 » pour fournir « /usr/bin/phar » (phar) en mode automatique
update-alternatives: utilisation de « /usr/bin/phar.php7.4 » pour fournir « /usr/bin/phar.phar » (phar.phar) en mode automatique


Creating config file /etc/php/7.4/cli/php.ini with new version
Paramétrage de apache2-bin (2.4.54-1~deb11u1) ...
Paramétrage de libhtml-template-perl (2.97-1.1) ...
Paramétrage de mariadb-server-10.5 (1:10.5.18-0+deb11u1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /lib/systemd/system/mariadb.service.
Paramétrage de mariadb-server (1:10.5.18-0+deb11u1) ...
Paramétrage de libapache2-mod-php7.4 (7.4.33-1+deb11u3) ...
Package apache2 is not configured yet. Will defer actions by package libapache2-mod-php7.4.

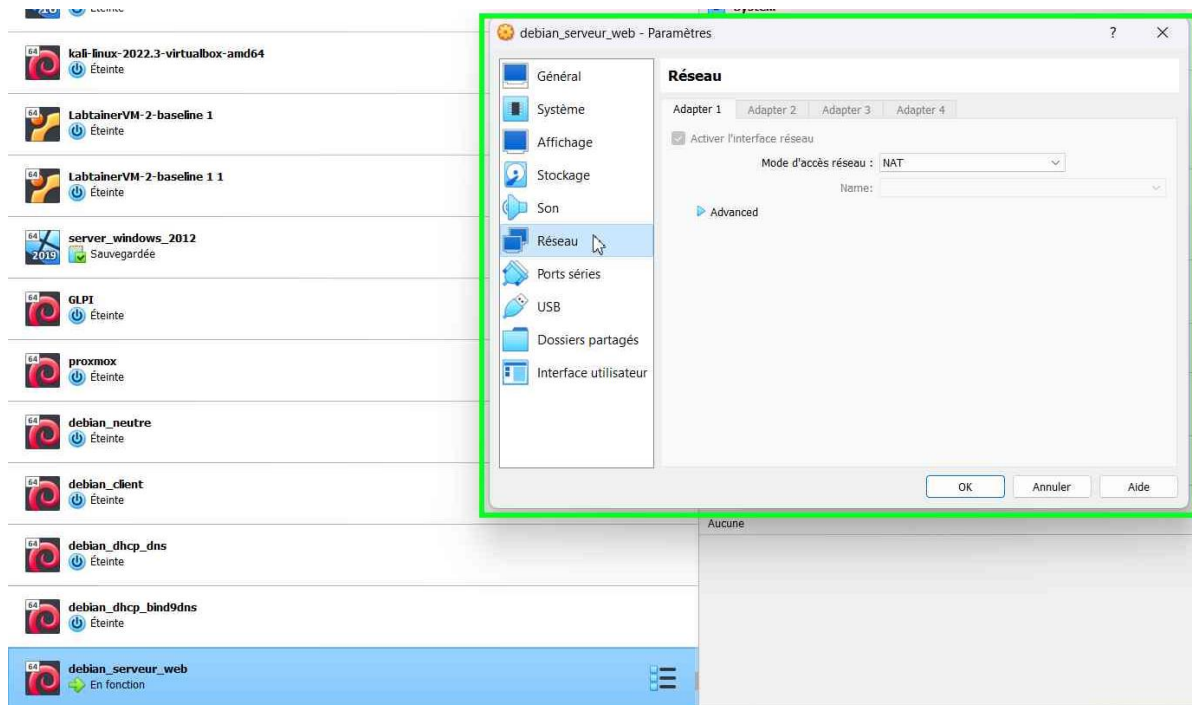
Creating config file /etc/php/7.4/apache2/php.ini with new version
No module matches
Paramétrage de libcgi-fast-perl (1:2.15-1) ...
Paramétrage de apache2 (2.4.54-1~deb11u1) ...

Progression : [ 94%] [#####.....]
```

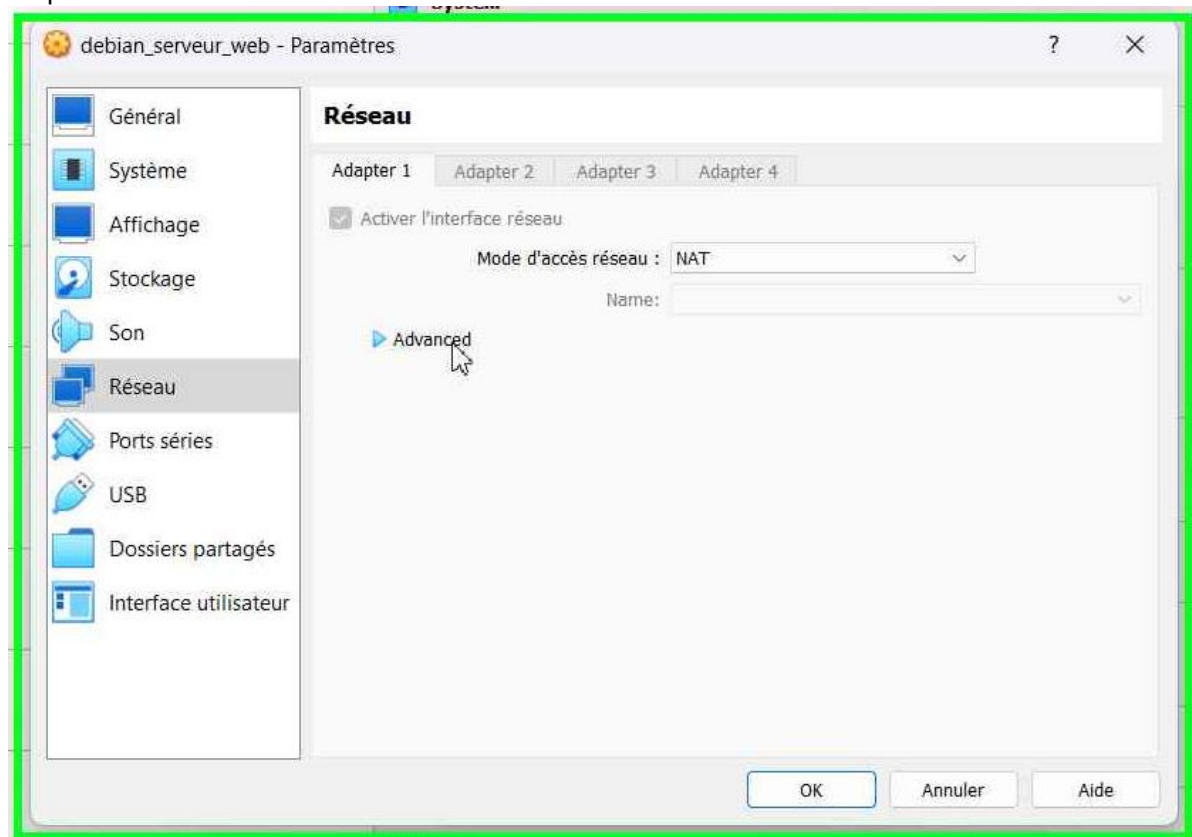
L'installation des paquets est terminée lorsque l'écran propose d'entrer une nouvelle ligne de commande.

Le réseau de la VM étant uniquement pour celle-ci (« NAT »), une redirection des ports est nécessaire.

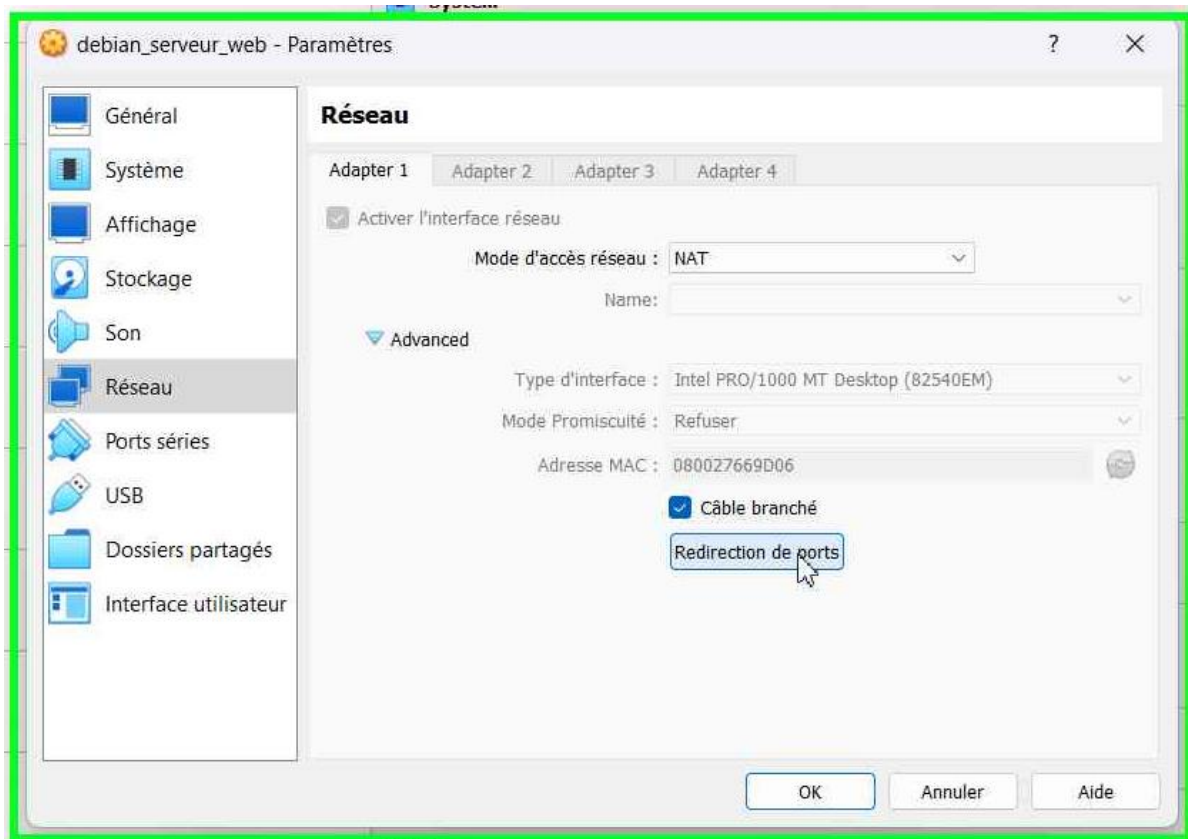
Vous devez sélectionner la VM et cliquer sur  puis « Réseau ».



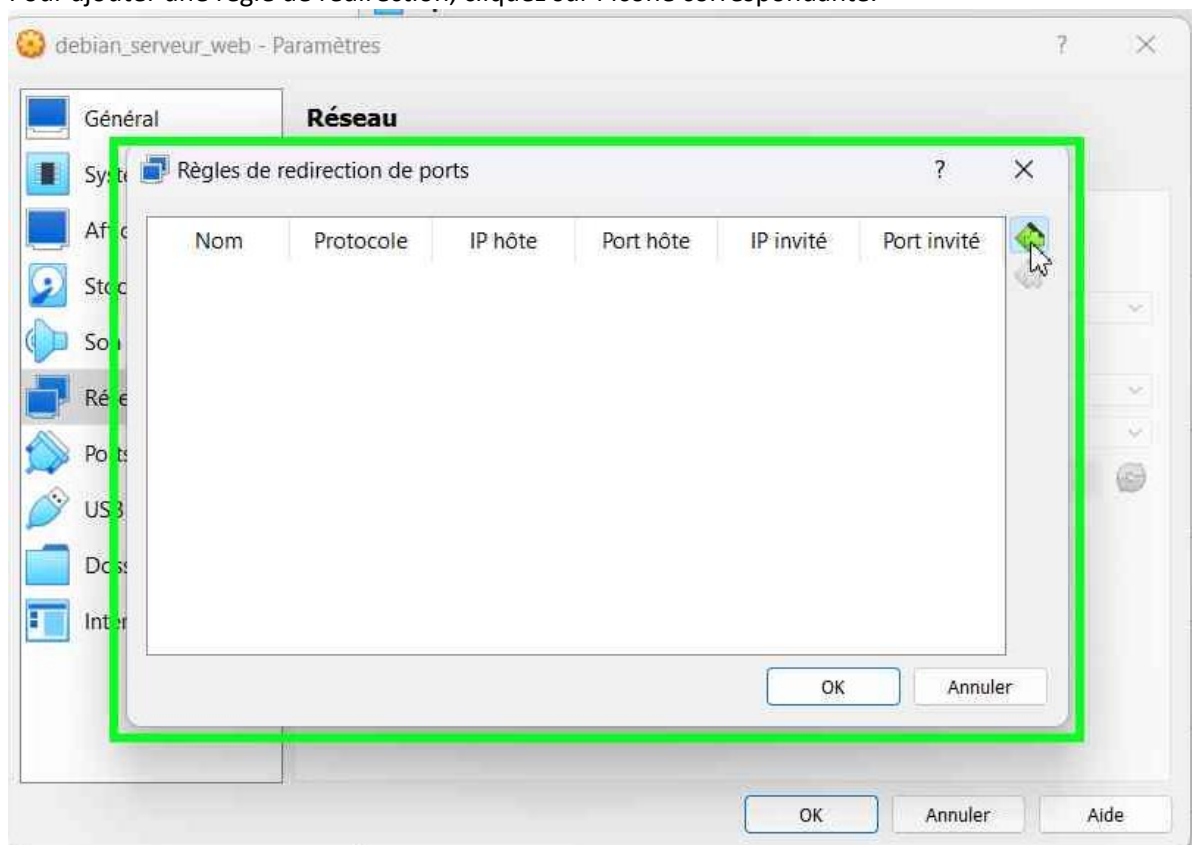
Cliquez sur « Advanced ».



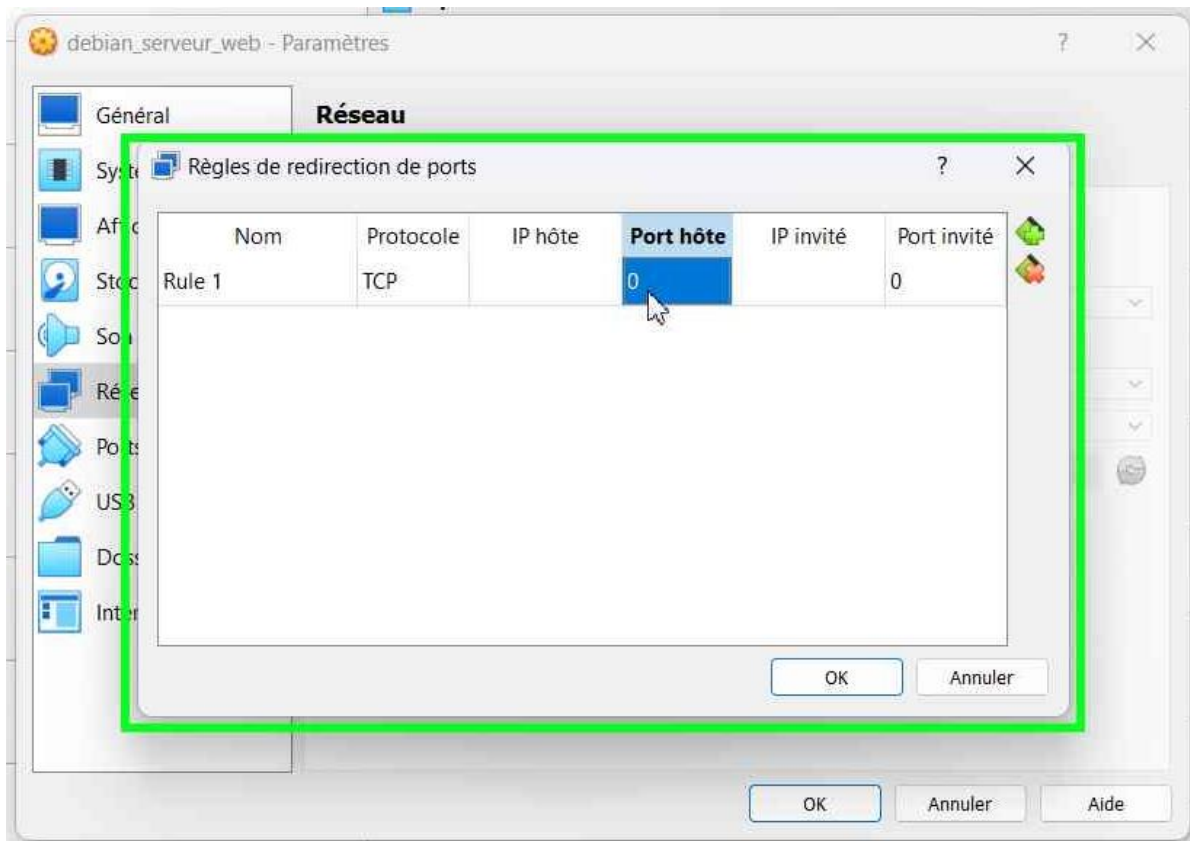
Puis, « Redirection de ports ».



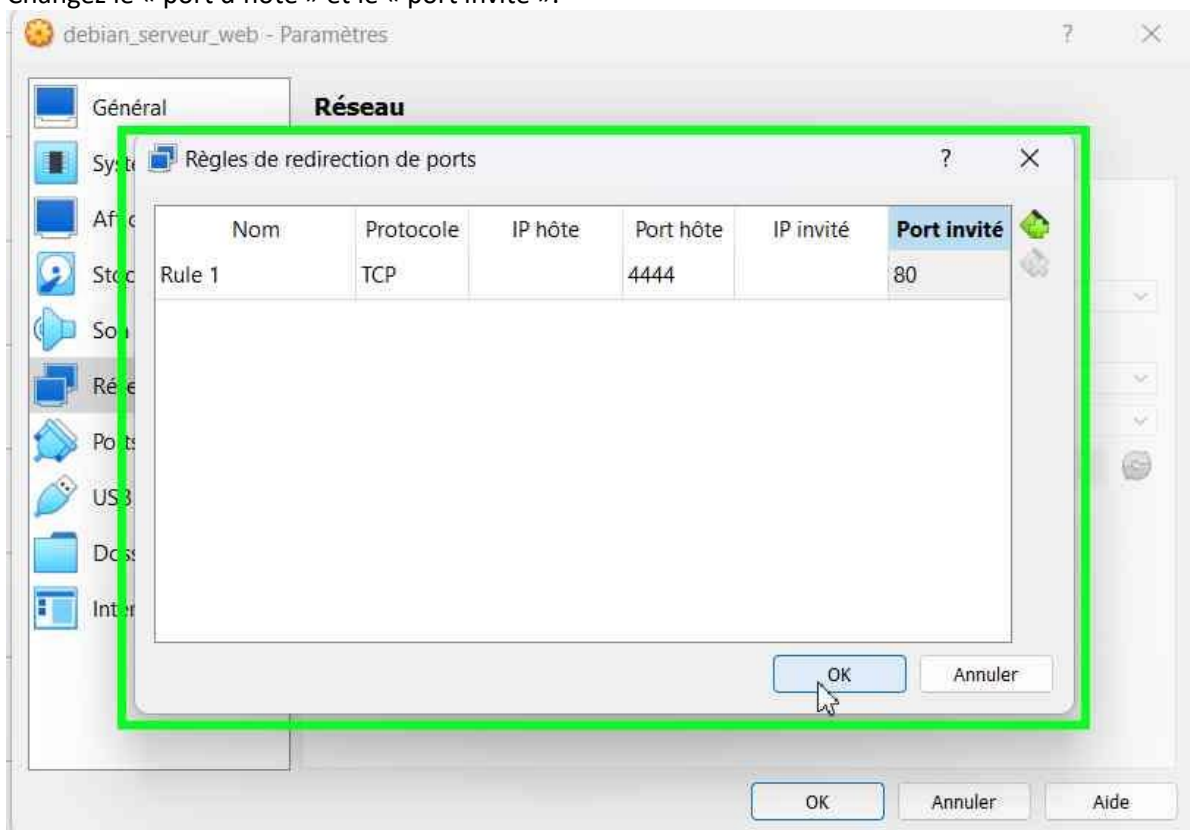
Pour ajouter une règle de redirection, cliquez sur l'icône correspondante.







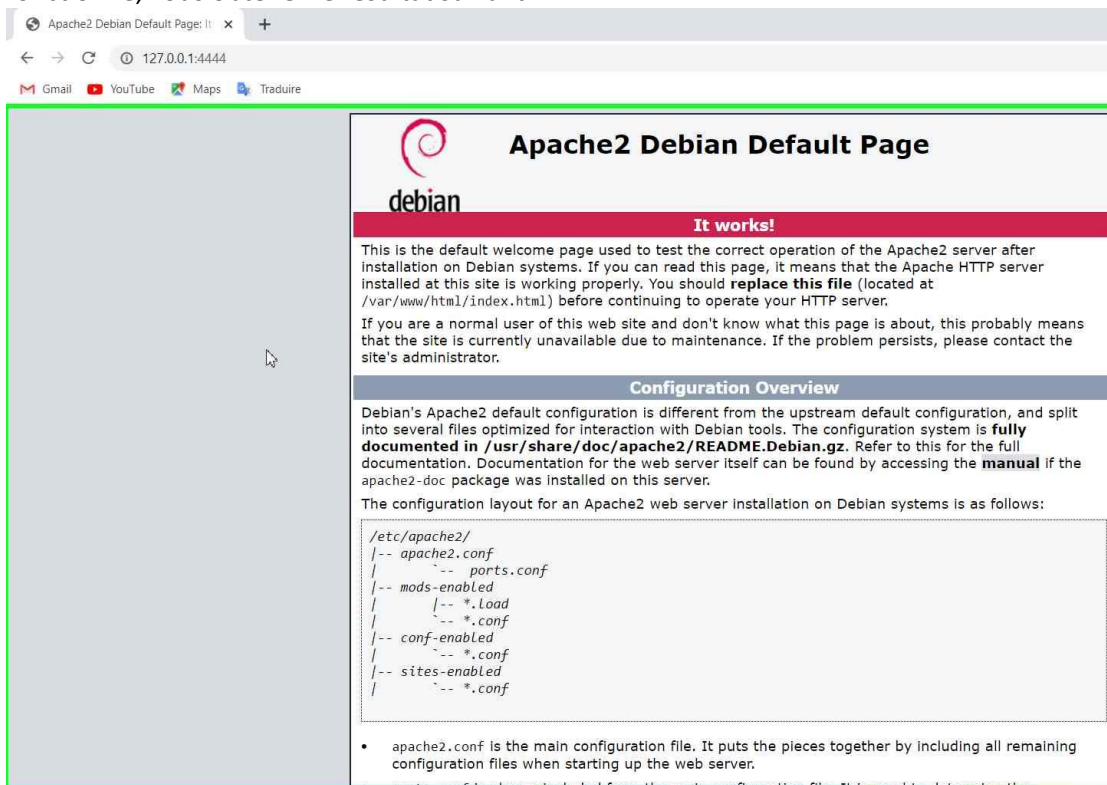
Changez le « port d'hôte » et le « port invité ».



Validez ces changements en cliquant sur « OK ».

Vous devez vérifier que la redirection des ports fonctionne.

Ouvrez un moteur de recherche et tapez : 127.0.0.1:[numéro de port d'hôte modifié]. Si la redirection fonctionne, vous obtenez le résultat suivant :



## 4- Fonctionnement

### a. Création de page HTML

Nous allons changer cette page de redirection en créant une page « test.html ».

Vous allez taper les commandes suivantes :

```
root@debian:/home/user# cd /var/www/html
root@debian:/var/www/html# ls
index.html
root@debian:/var/www/html# rm index.html
root@debian:/var/www/html# ls
root@debian:/var/www/html# nano test.html_
```

La première ligne de commande permet d'aller dans le dossier « html ».

La deuxième permet d'afficher le contenu de ce dossier.

La troisième permet de supprimer le contenu du dossier.

La quatrième permet de vérifier que le dossier est bien vide.

La cinquième permet de créer « test.html » et s'ouvre automatiquement :

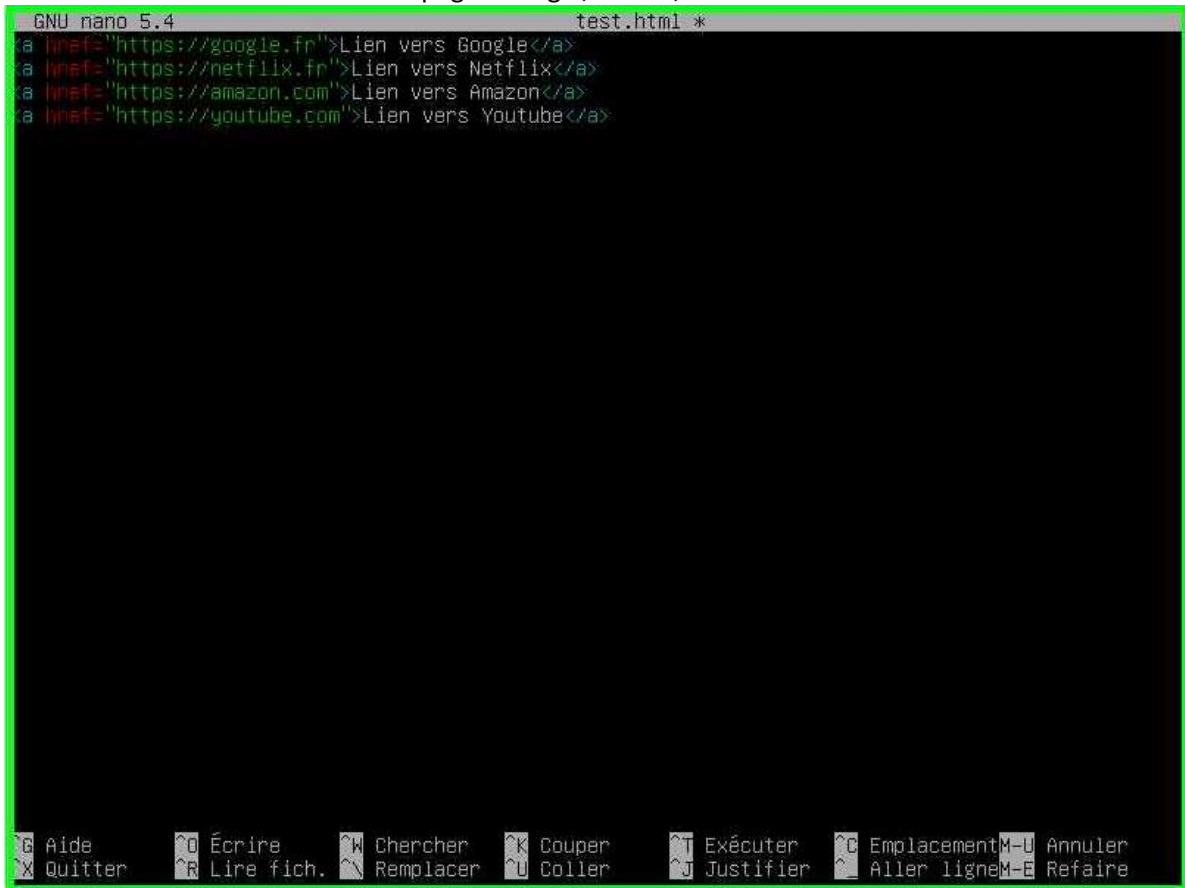




A ce stade, si « test.html » est fermé alors qu'il est vide, il ne sera pas enregistré.

On peut coder notre site avec HTML, CSS, PHP...

Nous allons créer des liens vers les pages Google, Netflix, Amazon et Youtube.



```
GNU nano 5.4 test.html *
^A href="https://google.fr">Lien vers Google</a>
^A href="https://netflix.fr">Lien vers Netflix</a>
^A href="https://amazon.com">Lien vers Amazon</a>
^A href="https://youtube.com">Lien vers Youtube</a>
```

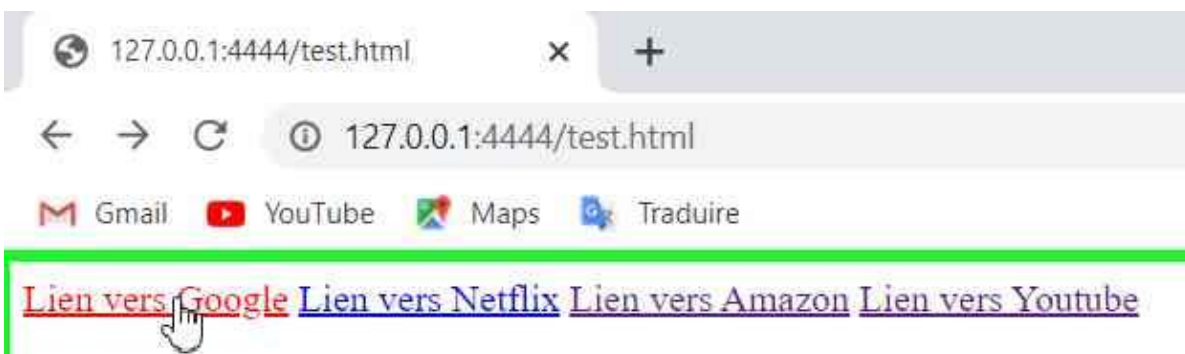
La combinaison de touches « Ctrl+O » permet de sauvegarder le contenu et « Ctrl+X » de fermer le fichier.

## b. Connexion à distance

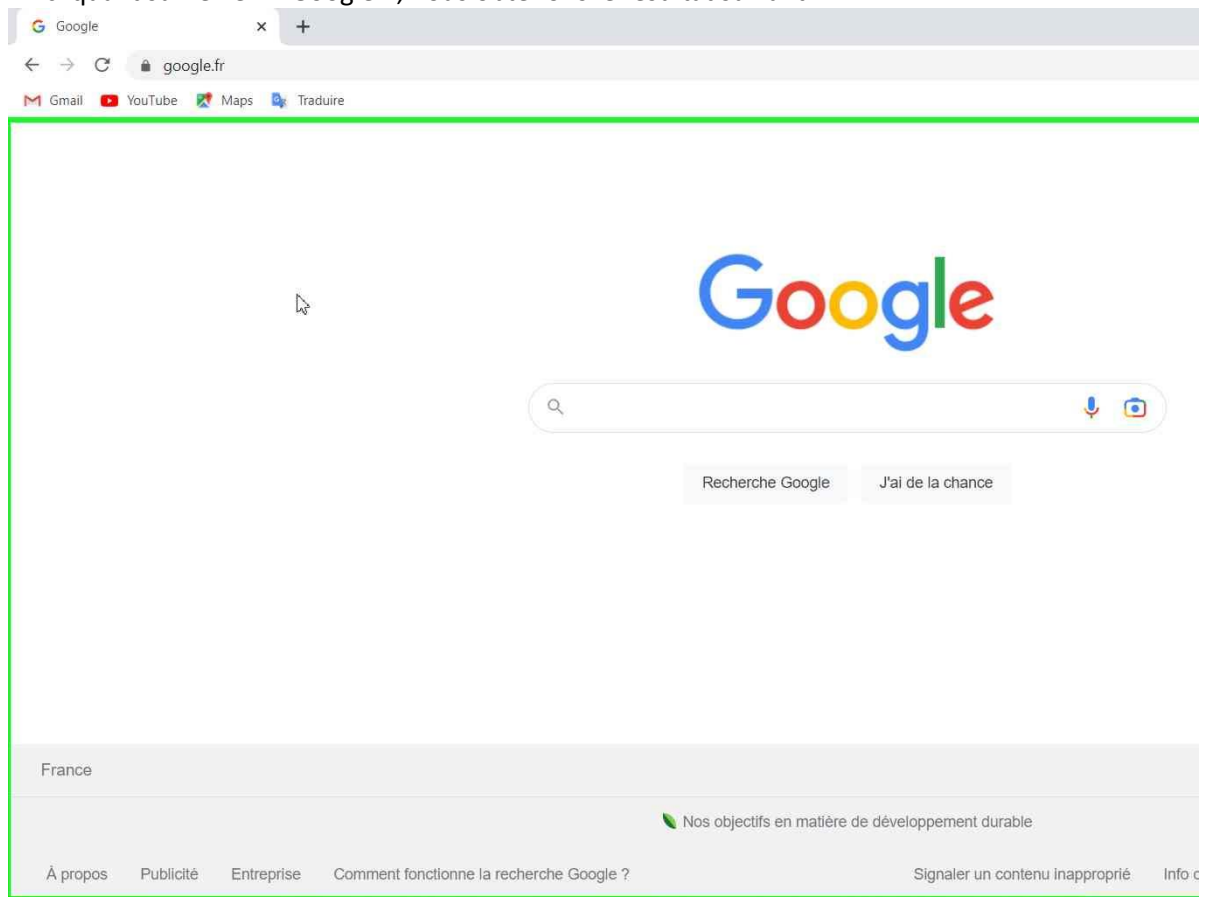
Nous devons vérifier que le fichier et le renvoi fonctionnent correctement.

Tapez « 127.0.0.1:[numéro de port d'hôte modifié]/test.html » dans la barre URL du moteur de recherche.

Attention : après le numéro de port d'hôte, une page intitulée « index.html » est recherchée par défaut. Si votre nouvelle page html est nommée autrement, il est nécessaire de préciser le nom complet de la page.



En cliquant sur le lien « Google », nous obtenons le résultat suivant :



Le serveur est terminé et fonctionnel.